Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Susanna Värtinen

Generating Role-Playing Game Quest Descriptions With the GPT-2 Language Model

Master's Thesis
Espoo, December 10th, 2021

Supervisor:	Professor Perttu Hämäläinen, Aalto University
Advisor:	Doctor Christian Guckelsberger

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Susanna Värtinen		
Title: Generating Role-Playing Game Quest Descriptions With the GPT-2 Language Model			
Date:	December 10th, 2021	**Pages:**	50
Major:	Computer Science	**Code:**	SCI3042
Supervisor:	Professor Perttu Hämäläinen		
Advisor:	Doctor Christian Guckelsberger		

Recent advances in artificial intelligence research have brought forth text-generating language models with promising computational storytelling capabilities. This thesis leveraged one of the most successful Transformer models, GPT-2, to automatically generate video game quest descriptions. We aimed to replace traditional procedural content generation methods for quests in games, as these often produce uninteresting, mechanical descriptions. We have gathered and processed a novel quest data set from a selection of popular 3D role-playing games. We have fine-tuned GPT-2 on the data set using various learning optimizations. Most notably, we replaced proper nouns within raw quest data with generic placeholder tokens to reduce unnecessary variance. We validated the resulting Quest-GPT-2 model via an online user study performed by role-playing game players. Our results indicate that one in five quest descriptions would be deemed acceptable by a human critic, yet the variation in quality across individual quests is large. Further work utilizing next-generation language models and our quest data set is expected to result in improved quest description quality.

Keywords:	artificial intelligence, generative models, procedural content generation, computational storytelling, quests, video games
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tieto-, tietoliikenne- ja informaatiotekniikan maisteriohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Susanna Värtinen		
Työn nimi: Roolipelitehtävien kuvausten generointi GPT-2-kielimallilla			
Päiväys:	10. joulukuuta 2021	**Sivumäärä:**	50
Pääaine:	Tietotekniikka	**Koodi:**	SCI3042
Valvoja:	Professori Perttu Hämäläinen		
Ohjaaja:	Tohtori Christian Guckelsberger		

Tekoälytutkimuksen viimeaikaiset edistysaskeleet ovat synnyttäneet tekstiä generoivia kielimalleja, joiden kyvyt algoritmisen tarinankerronnan saralla ovat lupaavia. Tämä diplomityö hyödynsi GPT-2:ta, yhtä menestyneintä Transformermallia, videopelitehtävien kuvausten automaattiseen generointiin. Tavoitteenamme oli korvata perinteiset videopelitehtävien luomiseen käytetyt proseduraaliset sisällönluontimetodit, sillä ne tuottavat usein mielenkiinnottomia, mekaanisia tehtävänkuvauksia. Olemme keränneet ja käsitelleet uudenlaisen pelitehtävistä koostuvan tietoaineiston muutamista suosituista 3D-roolipeleistä. Hienosäädimme GPT-2:ta käyttäen kokoamaamme tietoaineistoa hyödyntäen samalla useita oppimisen optimointitapoja. Korvasimme erityisesti perusmuotoisen tehtäväaineiston sisältämät erisnimet yleisluontoisilla tekstipaikkamerkeillä vähentääksemme turhaa vaihtelevaisuutta. Validoimme tuottamamme Quest-GPT-2-mallin roolipelipelaajille suunnatulla internetkyselytutkimuksella. Tuloksemme osoittavat, että ihmiskriitikot hyväksyisivät viidenneksen tehtävänkuvauksista, joskin laadun vaihtelu yksittäisten tehtävien välillä on suurta. On oletettavaa, että tehtävänkuvausten laatua voitaisiin parantaa seuraavan sukupolven kielimalleja ja pelitehtävätietoainestoamme hyödyntäen.

Asiasanat:	tekoäly, generatiiviset mallit, proseduraalinen sisällönluonti, algoritminen tarinankerronta, tehtävät, videopelit
Kieli:	Englanti

Acknowledgments

Firstly, I want to thank my advisor Christian Guckelsberger for providing support and feedback on this very thesis over the course of the somewhat unusual, strange year shadowed by the SARS-CoV-2 pandemic. His many comments on my writing did not go to waste. I also want to thank my supervisor Perttu Hämäläinen for making this thesis project possible in the first place and for providing additional funding to further improve the project (and for his helpful advice despite his busyness). Moreover, I want to thank them both for giving me an opportunity to work on an actual research paper.

Another set of thanks goes to the members of the Aalto game AI research group: thank you for your help on the text generation setting mini-studies. The many RPG buffs who participated in my online survey shall not be forgotten either: my thanks for your support. A big shout-out to everyone who wrote detailed answers to the free-form questions; your insights on the quest descriptions were rather interesting to read.

Last but not least, I want to thank my mom and dad for letting me crash at home out in the beautiful shores of Lake Saimaa without a worry in the world for the bulk of the duration of this thesis project. My grandma shall also receive a thank you for her many warm hugs.

Espoo, December 10th, 2021

Susanna Värtinen

Abbreviations

AI	Artificial Intelligence
GaaS	Games as a Service
GPT	Generative Pre-trained Transformer
MMO	Massively Multiplayer Online
NPC	Non-Playable Character
RPG	Role-Playing Game

Contents

Chapter 1

Introduction

Role-Playing Games (RPGs) contain various tasks, commonly known as quests, for their players to accomplish. Such quests are typically explicit, often narrative-driven, challenge tasks which usually reward the player who completes them. The most intriguing quests nowadays are still written by people, but it becomes hard to satisfy players' ever-growing demand for game content, especially in open world games and those that adhere to the Games as a Service (GaaS) model and thus are particularly reliant on long-term player engagement.

However, the existing approaches to create quests and their descriptions procedurally are lacking; traditional procedural quest generators come up with formulaic, repetitive quest descriptions. Meanwhile, research on Artificial Intelligence (AI) has brought forth powerful, new text-generating language models with improved computational storytelling capabilities. One such model is Generative Pre-trained Transformer 2 (GPT-2), which is based on the Transformer architecture and has been shown to produce various types of realistic, human-like texts with unprecedented quality [Radford et al., 2019].

This thesis explores the possibilities of utilizing GPT-2 to automatically generate quest descriptions, i.e. short texts that explain the quest to the player from the perspective of a quest-giving Non-Playable Character (NPC). While previous work has mostly studied quest generation in text games, our work focuses on 3D games. We have validated our GPT-2 model variant both objectively with training and validation loss as well as conditional perplexity scores, and subjectively via an online user study.

Our contribution is threefold: (i) a novel quest data set, a publicly available data set of 978 quests from six different RPG games; (ii) Quest-GPT-2, a quest-description generating variant of GPT; and (iii) a technique for replacing proper nouns and numbers with generic placeholder tokens which reduces variance in learning and helps a language model to focus on relevant pieces of information. Our quest data set has been made available at `https://github.com/svartinen/rpg-quest-data-set`.

Chapter 2

Background

This chapter begins with an overview of the larger research context: the field of computational creativity. Next, we discuss some important pieces of previous work on procedural quest generators and determine what can be learned from them. Afterwards, we introduce GPT-2, and its text producing capabilities in greater detail.

2.1 Computational Creativity

According to the Veale et al. [2019], computational creativity is a multi-disciplinary field of research that tries to model, simulate or replicate human-like creativity via computational methods. It encompasses aspects of computer science, psychology, cognitive science, and social anthropology [Veale et al., 2019], among others fields, but it is also considered a direct sub-field of the AI branch of computer science [Colton and Wiggins, 2012].

Some of the earliest notable contributions to computational creativity have been attributed to Margaret Boden and her book *The Creative Mind* [1990]. She argued that AI can help us understand the human mind and intuition, leading us to comprehend how we conjure up new ideas: this is the cognitive-social perspective to computational creativity [Pérez y Pérez, 2018], the first of the two major perspectives on the field. The second major perspective is the engineering-mathematical perspective focusing on the construction of creative systems [Pérez y Pérez, 2018], which is more relevant to this thesis, as we are building an AI model capable of writing RPG quest descriptions.

Boden [1990] also proposed two different levels for creativity: psychological creativity (P-creativity) and historical creativity (H-creativity). P-creativity occurs, when a person comes up with an idea or a combination of ideas that they personally have never thought before. On contrast, H-creativity applies only when an idea is completely original: no one has ever thought of that idea before in the span of human history [Boden, 1990]. Moreover, Boden also defined two additional types of creativity that utilize the notation of AI search spaces: exploratory creativity, i.e. searching a conceptual space for new ideas with known rules, and transformational creativity, i.e. expanding the idea search to unseen areas of the conceptual space by formulating new rules for traversal.

However, there are still some serious issues concerning computational creativity even if there are some agreed-upon definitions for creativity itself. In this vein, Colton and Wiggins [2012] have identified two major issues related to the field. Firstly, they have stated that the iterative methods to write software that is given more and more freedom in its creative outputs are still undefined. Secondly, the two researchers also have noted that there should be a satisfying scientific way to evaluate the creative outputs of software. On the bright side, there are some known frameworks that attempt to remedy the latter issue: Lamb et al. [2018] have written a comprehensive overview on many such frameworks.

Computational creativity has been applied in many domains associated with creative people [Colton and Wiggins, 2012]. These include storytelling and video game design, which together form the foundation of this thesis. Liapis et al. [2014] were among the first researchers to explore computational creativity in games, arguing that games would provide an interesting, multi-faceted platform for computational creativity research, perhaps even becoming the "killer app" for the field. Liapis et al. [2014] justified this by presenting visuals, audio, narrative, game mechanics and rules, level architecture, and gameplay as video game building blocks that could be implemented with a creative AI. In the past few years, other researchers have designed AI agents for implementing some of these building blocks: for example, the various iterations of the ANGELINA game design system [Cook and Colton, 2014; Cook and Smith, 2015; Cook et al., 2016a,b] have been used to create fully procedural games, covering all game building blocks more or less extensively.

Combining games and computational creativity would not only benefit the researchers of the field: the combination could also serve the interests of both game developers and players. To this end, Ventura [2016] has argued that creative AIs could design both game content and full games, thus relieving the human-creator bottleneck, and facilitate personalized gaming experiences by making believable, complex NPCs and player-tailored content, thus enhancing player experience.

2.2 Procedural Quest Generators

Procedural quest generators are among the most interesting applications of computational creativity in video games. However, relatively few commercial games feature procedural quests, which implies that there some large, unsolved issues in existing procedural quest generators. The existing methods for creating procedural quests could be either inferior to human quest designers, or too computationally intensive for real-world applications, for example.

There have been multiple attempts at designing procedural quests: Calvin and Michael [2007] have made an experimental game, *Charbitat*, which is centered around procedurally generated key and lock puzzles, Pita et al. [2007] have constructed a framework for creating dynamically linked procedural quests for persistent worlds with multiple players, and Stocker and Alvin [2018] have designed algorithms for generating non-linear quests based on implementation-specific rules and natural language verbs and nouns. The most influential work on procedural RPG quests, however, is probably the quest generator prototype by Doran and Parberry [2011], which has been expanded upon by Breault et al. [2021] and their Creation Of Novel Adventure Narrative (CONAN) system. Moreover, the genetic algorithms approach by Soares de Lima et al. [2019] represents a more modern attempt at procedural quest generation, which differs from traditional quest generation algorithms that directly construct quest graphs. We first present the three aforementioned pieces of work in greater detail, and later relate them to the work at hand. Lastly, we also discuss existing work combining video game quests and language models.

Doran and Parberry [2011] analyzed the quest structure of popular Massively Multiplayer Online (MMO) RPGs for formulating their quest generator prototype. They observed that human-authored quests have identifiable, common structures. Based on these structures, they have identified nine distinct categories for the underlying motivations of quest-giving NPCs. These categories represent the most important concerns of NPCs: quests are supposed to address the concerns to appear intentional, as opposed to random. From most to least common:

- Conquest: forcefully dominating over others
- Equipment: procuring or fixing equipment
- Knowledge: gaining new information, regardless of means
- Protection: protecting others or possessions from enemies

- Serenity: upholding justice or peace
- Reputation: doing impressive deeds that make one more famous
- Wealth: selling merchandise or obtaining merchandise to sell
- Comfort: improving living conditions
- Ability: gaining or improving either skills or equipment related to skills

Doran and Parberry [2011] also have defined two to seven options for quest objectives per motivation. For instance, Conquest involves either (i) attacking an enemy or (ii) stealing something. Additionally, each objective can be broken into sequences of simple actions: (i) requires first (1) going to an enemy, and then (2) damaging that enemy; (ii) consists of (1) going to the objective item, (2) stealing it, (3) returning to the client NPC, and, finally, (4) giving them the item [Doran and Parberry, 2011].

Doran and Parberry's [2011] quest generator implementation (Prolog backend, Java frontend) began the quest creation process with a user-picked NPC motivation. Then, their generator chose a random quest objective from a list of possible objectives for the selected motivation and added its actions into a binary tree. To complexify the quests, random actions were repeatedly replaced with subquests that consist of the same actions as the original quest objective [Doran and Parberry, 2011].

Doran and Parberry [2011] only performed brief mathematical analysis on the their quest generator; they noted that their approach produced unique quests with a high probability, because the leaves in random trees are unique with a high probability as well. They concluded that more work should be done to properly assess the quality of the produced quests versus human-written ones.

Breault et al. [2021] have expanded upon Doran and Parberry's work with their CONAN system. They aimed to produce infinite, coherent quests with believable character motivations and many alternative courses of actions for players. CONAN needed detailed information on a game world to function: locations, NPCs and their weighted preferences for actions, monsters, items, and laws describing the inner logic of the game world [Breault et al., 2021]. Breault et al. [2021] used a set of planning agents, one for every NPC, as a central part of CONAN. They initiated each agent with a character-specific planning problem with its own domain and sets of constrains and goals. During quest generation, the system selected the plan with the lowest cost as the quest shown to the player [Breault et al., 2021].

Breault et al. [2021] also constructed a built-in classification module that was used to evaluate the diversity of the quests generated by CONAN. The module classifies quests into categories which correspond with the types of motivation presented by Doran and Parberry [Breault et al., 2021]. Breault et al. created two test worlds, one small and another large, and let CONAN generate around a thousand quests per an NPC for both worlds without player interference. Afterwards, they used the classification module to classify the quests and charted the distribution of different motivations. For the larger world, all quest motivations were present, whereas only some of them were present in the smaller world [Breault et al., 2021]. As a result, Breault et al. [2021] argued that the complexity of the generated quests is dependent on the complexity of the game world and the amount of world information given to CONAN. They concluded that the system can write quests that resemble human-written ones. They also noted that a large number of available quests, and choosing which quests, sub-quests and actions to perform in the game world, provides interesting entertainment for a human player.

Soares de Lima et al. [2019] approached quest generation with a different technique. They used genetic algorithms, which are search heuristics inspired by Charles Darwin's *survival of the fittest* concept. Togelius et al. [2011] have given an overview of these algorithms in game content generation. Soares de Lima et al.'s [2019] implementation of the algorithms uses a fitness function that rated the quests according to how well their plots resembled a desired story arc: the fitness function received a series of symbols denoting rises or falls of tension in the story arc, converted the symbols to a numerical representation, and compared the story arc to the desired plot by utilizing mean squared error. Soares de Lima et al. used the standard single-point crossover.

Soares de Lima et al. [2019] performed a small user study to evaluate the quests produced by their quest generator implementation. The implementation was tailored for their own zombie-themed 2D RPG [Soares de Lima et al., 2019]. The researchers generated three quests for the game with their algorithm, and commissioned the same amount of quests from a professional game designer. Afterwards, they asked 34 students to play the game and complete its six quests. After completing a quest, the students decided whether the quest was designed by a human or an algorithm in a Turing-like fashion. Overall, the accuracy of the decisions was 49.02%, which is close to the random choice accuracy of 50%, meaning that the human-designed and algorithm-produced quests were practically indistinguishable [Soares de Lima et al., 2019].

On one hand, past implementations of procedural quest generators provide valuable knowledge about how to design an architecture for a quest generator. Doran and Parberry's [2011] work depicts different quest structures in great detail, and Breault et al.'s [2021] notion of character preferences supplements Doran and Parberry's work well. Furthermore, the genetic algorithms approach by Soares de Lima et al. [2019] could potentially be used to distill a small number of high-quality quests from a large collection of candidate quests.

On the other hand, there are issues related to the existing procedural quest generators that warrant future work. Firstly, Doran and Parberry's [2011] quest generator only produces linear quests: it cannot produce quests with alternative courses of action. This applies to Breault et al.'s [2021] CONAN as well. However, Stocker and Alvin [2018] have provided one possible solution to this problem. Secondly, Soares de Lima et al.'s [2019] approach is computationally taxing: computing a single quest takes a few minutes even with various optimizations. Thirdly, all three pieces of work could have been evaluated more exhaustively: Doran and Parberry [2011] only performed some mathematical analysis, Breault et al. [2021] made a small-scale simulation and used it to determine the capabilities of their quest generator, and the user study by Soares de Lima et al. [2019] had a small sample size and few quests to evaluate. In fact, most works on procedural quest generators seem to rely on simulations and comparisons to quests from existing games in quest evaluation instead of utilizing human players as evaluators even though video game quests are supposed to be experienced by people. Lastly, traditional quest generators do not concern themselves with detailed written narrative: standalone story generators exist (see the overview by Gervás [2009]), but they have mostly been used to create interactive stories, not video game quest descriptions.

Few researchers have explored the possibilities of applying language models to quest generation in text games. Firstly, Ammanabrolu et al. [2019] used GPT-2 for creating sequences of actions for quests. Secondly, Freiknecht and Effelsberg [2020] created an interactive story that used GPT-2 for writing the story and generating actions that could be taken at each turn. Thus, using a language model for writing quest descriptions in 3D games is not beyond the realms of possibility. 3D games do not need to describe environmental details, objects and characters as meticulously and thoroughly as text games, which in turn makes them more resistant to the limitations (e.g. forgetting finer details) of text-generating AI models. Altogether, 3D games have more layers of description, such as visuals and text, that can support each other, yet those layers should not overlap too greatly to avoid redundant information.

Table 2.1: Official OpenAI variants of GPT-2, named after their parameter counts

GPT-2 Variants
GPT-2-124M
GPT-2-355M
GPT-2-774M
GPT-2-1.5B

2.3 Generative Pre-trained Transformer 2

GPT-2 [Radford et al., 2019] is a text-generating language model by OpenAI, a company specialized on researching AI. The model has four official differently sized variants, as depicted in Table 2.1. It is the successor of GPT-1 [Radford et al., 2018] and the predecessor of GPT-3 [Brown et al., 2020].

All three iterations of GPT are based on the Transformer model [Radford et al., 2018, 2019; Brown et al., 2020]. The Transformer, designed by Vaswani et al. [2017], is an encoder-decoder language model relying on attention mechanics. In general, encoders turn variable length input data into fixed-sized feature maps, whereas decoders attempt to transform the maps back into assumed input. Attention mechanics, however, refer to evaluating the importance of each piece of input data. More specifically, the Transformer uses self-attention, where each input word is referred to other, possibly related words in the same input [Vaswani et al., 2017]. This approach establishes links between related words, such as names and pronouns.

Formally, each input word is translated into a vector embedding. Word embeddings are multiplied by three different weight matrices that are learned during training. Due to these matrices, here are three distinct representations of a word: a query, a value, and a key. Vaswani et al. [2017] defined the primary component of self-attention, i.e. the scaled dot-product attention shown in Figure 2.1, using these representations:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V,$$

where Q represents a query matrix, K denotes a key matrix, V represents a value matrix, and d_k is the dimension of the keys and values. Moreover, Vaswani et al. discovered that it was beneficial to project the queries, the keys, and the values linearly in parallel, as shown in Figure 2.1. This is

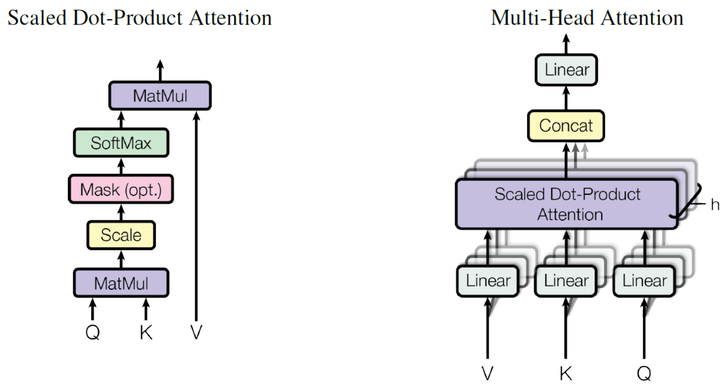

Figure 2.1: Self-attention (image from Vaswani et al. [2017])

reflected in the full self-attention definition

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_\mathrm{h})W^o,$$

$$\mathrm{head}_\mathrm{i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V),$$

where h is the number of parallel attention layers, and $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^o \in \mathbb{R}^{hd_v \times d_{model}}$.

GPT models only use decoder blocks unlike the original Transformer [Radford et al., 2018, 2019; Brown et al., 2020]. The GPT-2 version of this decoder-only architecture is depicted in Figure 2.2. Decoder-only architectures suit text-generating language models, because encoders and decoders are suspected of learning redundant information in mono-lingual language tasks [Liu et al., 2018]. As an additional benefit, Liu et al. [2018] state that decoder-only architectures can handle longer input-output pairs than full encoder-decoders.

The GPT family and other state-of-the-art language models differ greatly in their training procedures: the GPT models are trained with a diverse collection of unlabeled textual data and, optionally, fine-tuned with a small set of task-specific labeled training data afterwards [Radford et al., 2018; Vaswani et al., 2017]. Radford et al. [2018] named this type of unsupervised training *generative pre-training*. They also showed that the pre-training allows a language model to acquire a notable amount of common knowledge and establish long-range dependencies, or in other words, long-term memory. Pre-trained models can complete some specific tasks with good results even without fine-tuning, even though they perform better if they are given some task-specific fine-tuning [Radford et al., 2018].

GPT language models mostly differ from each other in terms of scale: GPT-2 has ten times more parameters than GPT-1, whereas GPT-3 has over

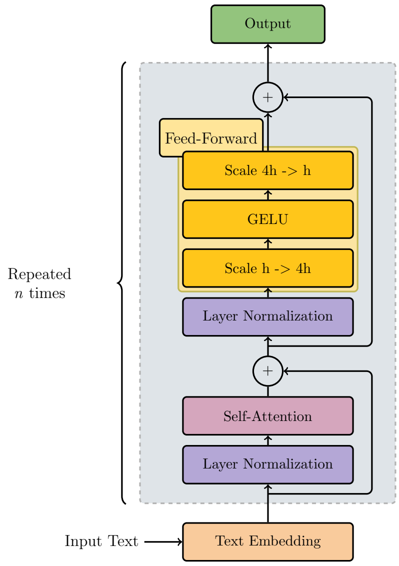

Figure 2.2: GPT-2 architecture (based on Radford et al. [2019]), variants of the model use either 12, 24, 36, or 48 as the value of n

one hundred times more parameters than GPT-2 [Radford et al., 2018, 2019; Brown et al., 2020]. The three models also had different pre-training data sets: GPT-1 was trained on the BookCorpus data set (7,000 unpublished books of various genres) [Radford et al., 2018], GPT-2 utilized the Web-Text data set (eight million web documents) [Radford et al., 2019], GPT-3 was given a compilation of Common Crawl (corpus of web data: raw web pages, metadata, text extracts), WebText2 (an expanded version of Web-Text), Books1 (a corpora of books), Books2 (another corpora of books) and Wikipedia (articles scraped from English Wikipedia) data sets (499 billion tokens, i.e. subwords, in total) [Brown et al., 2020].

Recalling computational creativity, scaling models and pre-training data sets larger and larger seems to provide at least a partial answer to Colton and Wiggins's [2012] first major issue, i.e. how to write more independently creative software iteratively. The larger GPT models do not need as extensive fine-tuning as the smaller ones. This is evident from the zero-shot learning capabilities of the GPT models: the larger the model, the more performant it is at zero-shot learning [Brown et al., 2020], i.e. learning from natural language instructions without explicit examples.

This thesis utilizes GPT-2 for various reasons. Firstly, the model has been tailored to text production based on input prompts, unlike GPT-1 [Radford et al., 2018, 2019]. Additionally, even smaller variants of GPT-2 should generate more coherent text than GPT-1 due to their larger pre-training data sets and minor architectural enhancements. Secondly, GPT-2 has been shown to generate more convincingly human-like sentences than various other language models, which makes it an attractive model to study even if it is not on the level of GPT-3. In comparison with GPT-3, GPT-2 is open-source, whereas GPT-3 is only available through the OpenAI API, making task-specific fine-tuning impossible for many researchers. Furthermore, OpenAI has only allowed select few researchers and commercial partners to access the API for limiting the misuse of the model [Brockman et al., 2020]. Even if the API was freely available, the running costs of using the API would be high, typically thousands of US dollars. For example, the costs of Philosopher AI have been estimated at 4,000USD per month [He, 2020].

GPT-2 has been used to produce various types of text successfully: poems that cannot be distinguished accurately from human-written ones [Köbis and Mossink, 2021], cooking recipes [Lee et al., 2020], and even a whole diary-like book by an AI writer that can be described highly paranoid by people [Agafonova et al., 2020]. These examples demonstrate the domain-agnostic nature of the model, thus providing credence to its creators' [Radford et al., 2019] claims of task generality. Moreover, the examples also display the computational creativity capabilities of the model; in particular, they show that GPT-2 can be used in writing tasks that can be considered creative by human observers.

Using Boden's [1990] definitions of creativity, it seems that GPT-2 can be used to produce text that is P-creative with a high probability when using a good word sampling strategy. A poor sampling strategy, however, leads to repetitive, unoriginal pieces of text. H-creativity of GPT-2, in contrast, is much harder to verify, but it is still plausible that the models might produce some unseen combination of ideas given its large pre-training data set. However, H-creativity diminishes over time and eventually reaches zero, if the model does not continually receive new training data: the creativity of the model is bound by the data it has seen.

Chapter 3

Training Data Set

While GPT-2 is known to be capable of producing good results in some text creation tasks out-of-the-box [Radford et al., 2019], the model requires some fine-tuning to generate quest descriptions. It is doubtful that the data set that was used to pre-train GPT-2 contains a sufficient amount of quest examples to facilitate high-quality quest generation without additional fine-tuning based on a separate, specialized data set.

As one contribution of this thesis, our quest data set is published along with the thesis for other researchers to adopt and potentially extend in their projects.

3.1 Collecting Data

Fine-tuning a language model can require a few thousand examples to produce good results, depending on the task and the size of the model. For instance, GPT-2-774M has been shown to require around 5,000 text samples, when fine-tuning the model for text continuation tasks [Ziegler et al., 2019]. Thus, collecting quests from multiple RPG games is necessary to obtain a large enough training set, assuming that quests cannot yet be created automatically with sufficient quality, or hand-authored within a sensible time-frame. Furthermore, RPGs from different game series are more likely to have distinct styles of quest writing, which in turn means that using examples from a set of games might allow a language model to learn more quest writing styles as well, provided that there are enough examples. Unlike regular RPGs, MMO games can have thousands of quests: *World of Warcraft* [Blizzard Entertainment, 2004] contains over 30,000 quests, for example. As a downside, quests in such MMOs are simpler in structure and less varied than their RPG counterparts: quests are mostly treated as busywork for player

Table 3.1: The quest data set (978 quests)

Game	**Sourcing**	**Quests**
Baldur's Gate [BioWare, 1998]	collected (game files)	100
Baldur's Gate II: Shadows of Amn [BioWare, 2000]	collected (game files)	94
The Elder Scrolls IV: Oblivion [Bethesda Game Studios, 2006]	collected (game files)	215
The Elder Scrolls V: Skyrim [Bethesda Game Studios, 2011]	collected (game files)	389
Minecraft [Mojang Studios, 2011]	written by the author	100
Torchlight II [Runic Games, 2012]	collected [van Stegeren and Theune, 2020]	80

character progression, as opposed to vehicles for role-playing or captivating story-heavy adventures. Thus, using only MMO quests in the training set would likely have a negative impact on the quality of the quests outputted by GPT-2.

There are two main techniques for obtaining video game texts: (i) extracting text directly from game files and (ii) scraping text from unofficial, fan-curated online sources, as noted by van Stegeren and Theune [2020]. However, video game text examples are hard to obtain: game files are often packed using poorly documented proprietary file formats if not encrypted, whereas fan-written sources, such as online wikis, typically only paraphrase the contents of the in-game texts, e.g. character dialogue, instead of directly documenting such texts as they appear to the players. Additionally, online wikis and other unofficial sources might contain quests that have been fabricated by the fans of the game.

Consequently, we (i) selected a few RPG games known for well-written quests, relatively high-quality fan wikis and active modding scenes, and (ii) extracted quest texts directly from the game files with modding tools while utilizing the quest sections of fan wikis as guides. In essence, high-quality fan wikis make it easier to find quest data within games files, whereas modding tools allow one to sidestep the issues related to file formats and encryption.

To arrive at a sufficiently large data set of varied and complex quests, we collected a total of 878 quest examples from several RPGs into our training set with the discussed quest collecting strategy, as depicted in Table 3.1. All in all, the listed games have medieval-esque fantasy settings: focusing on one setting should improve the quality of the model outputs, but it also limits the generative space of the model.

We gathered the quests in the following manner. Firstly, the quests from *Baldur's Gate I–II* were extracted by first identifying the quest-giving NPCs by reading the Baldur's Gate Wiki quest descriptions, then looking for and selecting the relevant game dialogue files with Near Infinity, a browser and

Table 3.2: Recognized quest ingredients

Quest Ingredient	**Description**	**Essential**
Quest-giver	The person giving the quest to the player	yes
Objective	The overarching goal of the quest	yes
Tasks	The actions that have to be done to fulfill the objective of the quest	yes
Task locations	The locations where the tasks can be completed	no
Rewards	The rewards given to the player upon the completion of the quest objective	no
Facts	Important facts related to the quest	no
Items	Important items related to the quest	no
Characters	Important characters related to the quest	no
Locations	Some secondary locations related to the quest	no
Groups	Important groups, e.g. factions, related to the quest	no
Enemies	Enemies that the player will face during the quest	no
Description	The quest description shown to the player; utilizes the other quest ingredients	yes

editor software for games that use the Infinity game engine, and finally using the relevant pieces of dialogue to construct proper quest descriptions. Secondly, the skeletons for *The Elder Scrolls IV–V* quests were first scraped from the Unofficial Elder Scrolls Pages in JSON format: each quest contained information on objective, locations, quest giver, and reward. The final quest descriptions were then formulated by reading the relevant game files with either The Elder Scrolls Construction Set (*The Elder Scrolls IV*) or the Creation Kit (*The Elder Scrolls V*). Lastly, the *Torchlight II* quests originally collected by van Stegeren and Theune [2020] were in .csv format with the following fields: speaker (quest-giver), text, dialogue type, quest name as seen in-game, quest name in game data, quest file, speaker unit type, unit file, and raw quest text. We converted these quests to our JSON schema (Appendix A), cleaned them up, and added any missing, relevant information, such as archetypal character descriptions.

Furthermore, we augmented the quest data set with one hundred manually written *Minecraft* quests to experiment with the generalization capabilities of the model, as seen in Table 3.1. Having some example quests that captivate the interesting, unique aspects of *Minecraft* should allow the fine-tuned model to express those aspects as well. With the inclusion of *Minecraft* quests, the quest data set comprises 978 quests. This number of quests should be suitably large for fine-tuning GPT-2-1.5B (vs. GPT-774M and 5,000 examples as mentioned), because quest texts are comparatively long, and larger GPT-based models are better at few-shot learning [Brown et al., 2020], i.e. they learn patterns and trends from fewer examples.

```
<|begin_quest|>
<|begin_objective|>
kill Dynaheir
<|end_objective|>
<|begin_tasks|>
find Dynaheir
<|end_tasks|>
<|begin_task_locations|>
west of Nashkel, near
the gnoll stronghold
<|end_task_locations|>
<|begin_quest_giver|>
Edwin: a pompous wizard
<|end_quest_giver|>
<|begin_rewards|>
one year of Edwin's
services as a wizard
<|end_rewards|>
<|begin_characters|>
Dynaheir: a treacherous
female witch
<|end_characters|>
<|begin_locations|>
Nashkel: a town
<|end_locations|>
<|begin_tools|>
NONE
<|end_tools|>
<|begin_description|>
I am Edwin, a wizard,
and I require you!
(Yes, they will do
nicely.)
I would have you kill a
witch, the witch
Dynaheir. She is
treacherous, but with
your participation I
foresee no difficulty.
Last I heard of her,
she was traveling to
the west of Nashkel,
close to the gnoll
stronghold located
there. Will you assist?
The prize I offer would
surely be beyond
measure in your meager
understanding. Your
payment shall be one
year of my services as
a wizard. I am sure you
agree that my guidance
will be far more
valuable than any
monetary sum.
<|end_description|>
<|end_quest|>
```

(a) XML-like

```
This is an RPG quest
from Baldur's Gate.

Objective:
kill Dynaheir

Tasks:
find Dynaheir

Task locations:
west of Nashkel, near
the gnoll stronghold

Quest-giver:
Edwin, a pompous wizard

Rewards:
one year of Edwin's
services as a wizard

Characters:
Dynaheir: a treacherous
female witch

Locations:
Nashkel: a town

Quest description, the
quest-giver explaining
the quest to the player:
I am Edwin, a wizard,
and I require you! (Yes,
they will do nicely.)
I would have you kill a
witch, the witch
Dynaheir. She is
treacherous, but with
your participation I
foresee no difficulty.
Last I heard of her, she
was traveling to the
west of Nashkel, close
to the gnoll stronghold
located there. Will you
assist?
The prize I offer would
surely be beyond measure
in your meager
understanding. Your
payment shall be one
year of my services as a
wizard. I am sure you
agree that my guidance
will be far more
valuable than any
monetary sum.
```

(b) Simple

```
This is an RPG quest from
Baldur's Gate.
The quest-giver is called
Edwin. Edwin is a pompous
wizard.
The quest-giver gives a
quest to the player. The
player's objective is to
kill Dynaheir.
The player should first find
Dynaheir to complete their
objective. This task can be
completed in the following
location: west of Nashkel,
near the gnoll stronghold.
The player will receive the
following rewards for
completing the quest
objective: one year of
Edwin's services as a wizard.
The following characters are
related to this quest:
Dynaheir (a treacherous
female witch).
The following locations are
related to this quest:
Nashkel (a town).
This is the quest
description, the quest-giver
explaining the quest to the
player:
"I am Edwin, a wizard, and I
require you! (Yes, they will
do nicely.)
I would have you kill a
witch, the witch Dynaheir.
She is treacherous, but with
your participation I foresee
no difficulty. Last I heard
of her, she was traveling to
the west of Nashkel, close
to the gnoll stronghold
located there. Will you
assist?
The prize I offer would
surely be beyond measure in
your meager understanding.
Your payment shall be one
year of my services as a
wizard. I am sure you agree
that my guidance will be far
more valuable than any
monetary sum."
```

(c) Narrative

Figure 3.1: An example quest in the three proposed quest metadata formats

3.2 Data Formatting

To generate a quest description, a language model must first be given an outline depicting the various vital, general "ingredients" of a quest. We analyzed the collected quests to recognize these ingredients: the results of the analysis are shown in Table 3.2. Our quest ingredients align partially with classical narrative analyses in literature, such as Vladimir Propp's *Morphol-*

ogy of the Folktale [1968]. For example, Propp's definitions of various types of dispatchers, i.e. quest-givers, and character archetypes bear similarities to ours. Unfortunately, most narrative analyses approach quests as functions of events from the beginning to the end, whereas we are more interested in depicting the beginning of a quest in great detail.

The final piece in the puzzle is the training data format. The manner in which information is laid out to a language model is crucial: semantically equivalent pieces of input text might yield wildly different results, because some text formats likely synergize better with the pre-training data of the model than others. We devised three distinct textual formats, i.e. quest metadata formats, for representing the quests via their quest ingredients: a highly structured format that resembles XML, later referred to as *XML-like*, a *simple* format that is inspired by *dramatis personae*, i.e. character listings in plays and movie scripts, and, finally, a format that reads like a small story, dubbed *narrative*. Fig. 3.1 depicts an example quest, *Edwin and Dynaheir* from *Baldur's Gate*, in these three formats. The first format, *XML-like* is adopted from the work of Lee [2020], who has successfully used a similar structured metadata based format to generate patent claims with GPT-2.

We also devised a generic JSON representation for storing our quests in an organized manner (Appendix A). This representation was used to derive textual training data in the three discussed quest metadata formats. Storing quests in a well-known format also enables other researchers to adopt, parse and potentially convert our collection of quests.

3.3 Data Processing

While collecting the quest data set, candidate quests were evaluated by the author based on the following criteria:

- The novelty and interestingness of narrative and content [Gervás, 2009].
- The existence of clearly defined goals.
- The length of the quest description: too short descriptions might not contain all required quest ingredients, whereas too long descriptions might exceed the context window of GPT-2, thus resulting in the model forgetting some of the quest ingredients.

Some candidates did not meet one or multiple criteria and were consequently omitted. Other quests only met these criteria to a limited extend, and were consequently manually edited. In greater detail, quests are usually delivered through sprawling dialogue between the player and the quest-giver, not linearly through monolithic pieces of text. As a consequence, details like quest rewards are commonly discussed after the player has already completed the quest; we had to make some tense changes to accommodate the rewards into the quest descriptions. Moreover, some candidate quests were broken into multiple quests in the quest data set: these candidates either (i) involved the quest-giver directing the player to another NPC, or (ii) had distinct paths for the player to follow depending on their in-game actions.

Chapter 4

Developing Quest-GPT-2

To fine-tune GPT-2 into a capable quest writer, Quest-GPT-2, using efficient quest formatting is of paramount importance. The text generation example in Fig. 4.1 shows that GPT-2 can generate some short, rudimentary quest descriptions even without fine-tuning, if one provides few quest examples for it in the input text. However, quest descriptions typically incorporate many small elements, such as world knowledge, as well as character relationships and archetypes. It is difficult to incorporate those elements into a few quest examples in the input, especially considering the fact that the context window, or short-term memory, of GPT-2 holds only 1,024 tokens, i.e. byte-pair encoded sets of characters.

```
objective: kill all creepers
location: woods
quest giver: a butcher
reward: a diamond axe
description: Creepers have taken over the woods! Hunters can't procure game for me!
Kill all creepers! I'll reward you with a diamond axe.

objective: save villagers from a witch
location: a village
quest giver: a villager
reward: 16 emeralds
description: A witch is holding my fellow villagers captive. Someone ought to save
them! Traveler, if you did this task for me, I'd give you 16 emeralds.

objective: kill all zombies
location: caves
quest giver: a villager
reward: 32 golden carrots
description: Zombies are out for blood! Kill all zombies! I'll reward you with 32 golden carrots.
```

Figure 4.1: A simple quest generation example with (not fine-tuned) GPT-2-774M, when providing some quests as input (output in bold)

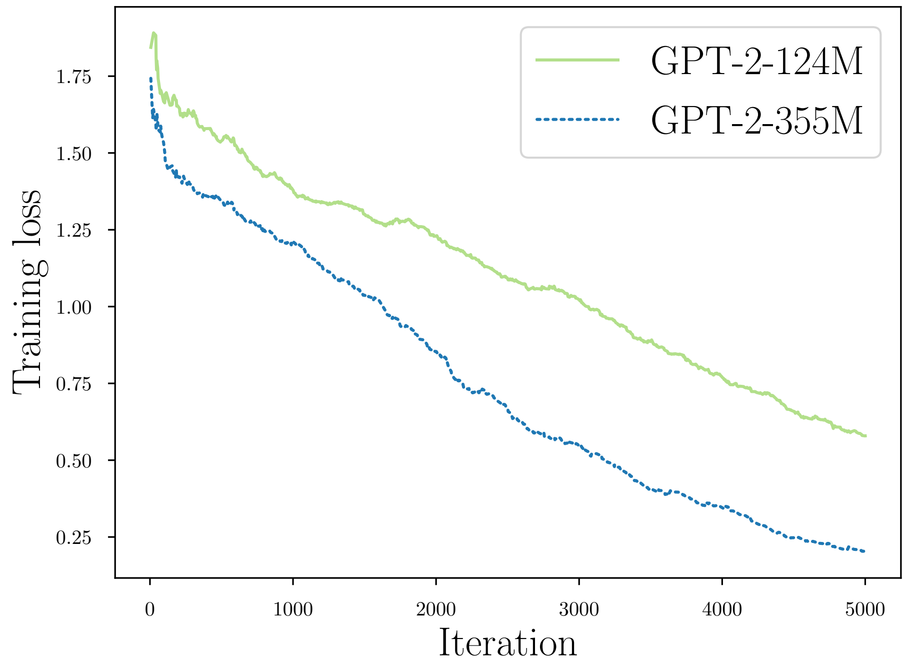

Figure 4.2: Early fine-tuning experiments with the quests shown in Table 3.1, moving averages of cross-entropy loss

4.1 Preliminary Fine-Tuning Experiments

We began the fine-tuning process with a series of quick, small experiments on an Nvidia GTX 1070 8GB GPU with the two smallest model variants of GPT-2: GPT-124M and GPT-355M. We utilized the *XML-like* quest metadata format in these early experiments. We fine-tuned the two models by using the training script from @nshepperd's fork of the official OpenAI GPT-2 Github release. We used @nshepperd's default optimizer settings, i.e. the Adam optimizer with an initial learning rate of $2 * 10^{-5}$, and a batch size of one, because larger batch sizes generated out-of-memory exceptions with 8GB of VRAM. Fig. 4.2 shows the average training losses for the last initial experiment. Additionally, Fig. 4.3 demonstrates generating quest descriptions with the fine-tuned models from Fig. 4.2. Fig. 4.3a depicts an input quest outline, i.e. quest ingredients in a certain quest metadata format sans the text for the quest description, and Fig. 4.3b and Fig. 4.3c show corresponding randomly selected output quest descriptions.

We made some small observations in-between adjustments to and repetitions of this setup. Firstly, if the characters have not been explicitly gen-

```
<|begin_quest|>
<|begin_objective|>
kill Zandalar
<|end_objective|>
<|begin_tasks|>
find Zandalar
<|end_tasks|>
<|begin_task_locations|>
Glintwood: a sparkling
fairy forest
<|end_task_locations|>
<|begin_quest_giver|>
Emmeth: a guard with a
strong sense of justice
<|end_quest_giver|>
<|begin_rewards|>
1,500 platinum chips
<|end_rewards|>
<|begin_characters|>
Zandalar: a male wizard
and a dangerous lunatic
<|end_characters|>
<|begin_tools|>
Zandalar has killed
several people before
<|end_tools|>
<|begin_description|>
```

(a) Input quest outline in the *XML-like* format

```
Oh my! You've come for me, wizard! You kill so
many people, and yet you still have hunger! No
hunger! There's still blood! Forgive me! I had a
sneaking suspicion... some of the guards would
startle Zandalar. Didn't expect him, but he was a
brutal wizard.
You'll need to find Zandalar. Help a guard. Make a
delivery to Zandalar. Then use the time between
each death to kill his cronies. Make the right
choice.
<|end_description|>
<|end_quest|>
```

(b) GPT-2-124M output quest description

```
Someone has marked our vigil. What do you want?
Someone has apparently gone through the large gate
into the Glintwood and killed several guards. You
must go after them and find the culprit. There is
money to be had, as well as the reputation of a
danger to yourself and others.
I will pay you 1,500 platinum, plus a portion of
the cost of the reward I have offered.
<|end_description|>
<|end_quest|>
```

(c) GPT-2-355M output quest description

Figure 4.3: A quest generation example with the models from Fig. 4.2, randomly selected examples generated with aitextgen using its default generation settings

dered in the metadata, both employed variants of GPT-2 might either choose a binary gender, or randomly flip between male or female pronouns. This behavior was fixed by explicitly including the characters' genders in their descriptions in later experiments. Secondly, both models displayed signs of over-fitting in all experiments: using a strategy like early stopping should improve the subjective quality of generated quest descriptions.

On one hand, these early experiments showed promise for generating relatively coherent quest descriptions, and even complete quests, when supplying the <|begin_quest|> token as input. On the other hand, the generated descriptions do not always encompass all quest ingredients of the input quest outline, and entities might be treated incorrectly. Most strikingly, a character who is referenced multiple times in a quest outline might appear as several separate people in the output quest description. When comparing the two differently sized GPT-2 variants, the larger GPT-2-355M produced noticeably more coherent quest descriptions than the smaller GPT-2-124M, while also transmitting the ingredients of the input quest outlines into output quest descriptions more comprehensively. Additionally, the cross-entropy loss for the larger GPT-2-355M converges noticeably faster towards zero than the loss for the smaller GPT-2-124M.

```
<|begin_quest|>
<|begin_objective|>
kill character_0
<|end_objective|>
<|begin_tasks|>
find character_0
<|end_tasks|>
<|begin_task_locations|>
west of location_0 , near the gnoll stronghold
<|end_task_locations|>
<|begin_quest_giver|>
quest_giver: a pompous wizard
<|end_quest_giver|>
<|begin_rewards|>
one year of quest_giver 's services as a wizard
<|end_rewards|>
<|begin_characters|>
character_0: a treacherous female witch
<|end_characters|>
<|begin_locations|>
location_0: a town
<|end_locations|>
<|begin_tools|>
NONE
<|end_tools|>
<|begin_description|>
I am quest_giver , a wizard , and I require you! (Yes, they will do nicely.)
I would have you kill a witch, the witch character_0. She is treacherous, but
with your participation I foresee no difficulty. Last I heard of her, she was
traveling to the west of location_0 , close to the gnoll stronghold located
there. Will you assist?
The prize I offer would surely be beyond measure in your meager understanding.
Your payment shall be one year of my services as a wizard. I am sure you agree
that my guidance will be far more valuable than any monetary sum.
<|end_description|>
<|end_quest|>
```

Figure 4.4: An example quest in the *XML-like* format with *placeholder text*

4.2 Substituting Proper Nouns and Numbers With Placeholders

To address the issues that came up in the early experiments, we decided to introduce two different strategies for deriving the textual quest metadata formats in the final fine-tuning: using original *raw text*, as depicted in Fig. 3.1, and secondly, utilizing *placeholder text* where proper nouns, i.e. unique names, and numbers are replaced with post-processable placeholder tokens. Fig. 4.4 displays the example quest from Fig. 3.1 in *XML-like* format with placeholders. We hypothesized that the placeholders should make the quest examples more palatable for GPT-2, because it allows the model to learn meaningful content independently from the name and number information that bears no significant meaning.

In theoretical terms, generative models like GPT-2 learn complex multivariate probability densities $p(x, y, ...)$: learning these densities becomes more difficult as the number of variables grows. However, we can assume that names are independent from other quest content, and that the joint distribution can thus be factorized into $p(x, y, ...) = p(x)p(y, ...)$.

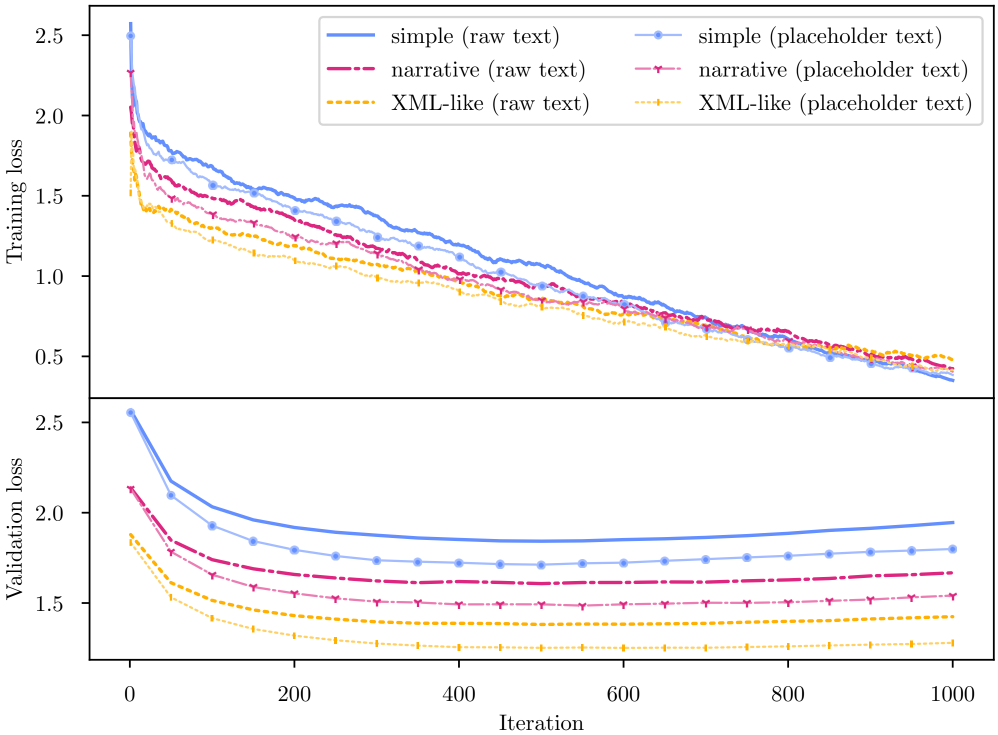

Figure 4.5: Fine-tuning results, moving averages of cross-entropy loss

4.3 Fine-tuning Quest-GPT-2

We split the training data, i.e. the 978 quests from Table 3.1, into training, validation, and test sets with 80:15:5 percent ratios. We used the validation set to combat over-fitting, and the test set for evaluation against human judgment in user studies. More specifically, the quests were split into the three sets randomly per game, and then combined into the complete training, validation, and test sets: this approach guaranteed that all games were represented in roughly equal proportions in all sets. Afterwards, we converted the sets into the three proposed quest metadata formats, producing both *raw text* and *placeholder text* for each format.

In contrast to the preliminary experiments, we fine-tuned the full-sized 1.5B parameter GPT-2 model, i.e. GPT-2-1.5B. We repeated training six times, once for each combination of metadata format and text type, i.e. *raw text* and *placeholder text*. We fine-tuned the model for 1,000 iterations at most and stopped early if the current validation loss was the minimum validation loss encountered so far. We used the same fine-tuning settings as in

```
This is an RPG quest from
a medieval fantasy video
game.
The quest-giver is called
quest_giver. quest_giver
is a guard with a strong
sense of justice.
The quest-giver gives a
quest to the player. The
player's objective is to
kill character_0.
The player should first
find kill character_0 to
complete their objective.
This task can be completed
in the following location:
task_location_0 (a
sparkling fairy forest).
The player will receive
the following rewards for
completing the quest
objective: number_0
platinum chips.
The following characters
are related to this quest:
character_0 (a male wizard
and a dangerous lunatic).
There are some important
facts concerning this
quest. character_0 has
killed several people
before.
This is the quest
description, the
quest-giver explaining the
quest to the player:
"
```

(a) Input quest outline in the *narrative* format with *placeholder text*

```
You're going to kill character_0. That man has
killed people before.
He is a dangerous lunatic. The only one that can
stop him now is character_0 himself.
Get up on that tower and kill character_0. Make sure
that he never gets out of this tower. Until he is
dead, none of this will ever happen.
```

```
Do you want to help us? quest_giver of Justice gives
you number_0 platinum chips! I need to get rid of
someone who has killed many of my family members. It
was some guy, character_0, a wizard. character_0
must burn in hell for his crimes.
You must help quest_giver. Do this quest and be
rewarded. Get rid of character_0, and you'll receive
number_0 platinum chips! Be quick on your feet!
```

```
What in blazes is happening in the task_location_0?
Someone's trying to murder people right under our
noses!
They've been killing, they've been murdering in a
dangerous, mysterious, unearthly force! There was a
group of wizards in that task_location_0. Now I
don't know, I don't even know what they were doing.
But they killed people so I could, I couldn't do
anything!
Well, one of them did something that was a bit more
tricky, let me tell you what. He wanted to become a
wizard, and he started to learn, and the wizards
tried to stop him. So the wizard became a wizard.
And now he's, I don't know what he is, a bad person,
but he's a very dangerous lunatic. No one knows
where he is. I want to kill him!
```

(b) Three random output quest descriptions generated with the final Quest-GPT-2 model

Figure 4.6: A quest generation example after the final fine-tuning, utilizes aitextgen and its default generation settings

the preliminary experiments in Section 4.1. The fine-tuning was performed on an Nvidia V100 32GB GPU (courtesy of Aalto University School of Science "Science-IT" project) and it took roughly 50 minutes for each metadata format and text type combination.

Fig. 4.5 shows the results of the fine-tuning. The trends in both cases are similar and the findings unanimous: *placeholder text* performs the best in terms of both training and validation loss for all metadata formats. In terms of metadata formats, the *XML-like* format achieves the smallest training and validation loss, whereas the *simple* format performs the worst. Yet comparing metadata formats only via loss values can be misleading: the model might learn repetitive formatting easily, thus "masking" the loss values smaller when using heavier formatting.

To rule this out, we employed perplexity, an established language model evaluation metric that measures how "surprised" a model is upon seeing a

Table 4.1: Conditional perplexities of the fine-tuned models

Metadata Format	Text Type	Conditional Perplexity
narrative	raw text	10.63
	placeholder text	**10.50**
simple	raw text	10.95
	placeholder text	10.55
XML-like	raw text	11.05
	placeholder text	10.78

piece of text. We calculated the conditional perplexities of the quest descriptions in the validation set, i.e. the perplexity of a quest description when given a certain input quest outline. These perplexities are depicted in Table 4.1; the table shows that *placeholder text* achieves lower perplexity than *raw text* with all three different metadata formats, thus supporting our previously discussed assumptions about *placeholder text*. Moreover, there is an observable trend in the ordering of the three metadata formats: *narrative* achieves the lowest perplexity regardless of the used metadata format variant, whereas *XML-like* always produces the highest perplexities.

Based on these insights, the model that was fine-tuned using the *narrative* format and *placeholder text* was chosen as the final Quest-GPT-2 model due to the subjective quality of its outputs, and its objectively lowest conditional perplexity score.

4.4 Exploring Quest-GPT-2 Text Generation Settings

We anticipate that many generated quests after the fine-tuning would still not convince a human audience. For example, Fig. 4.6 shows quest descriptions that can be considered somewhat nonsensical by human readers. On a brighter note, it is known that sampling methods, such as top-k sampling and nucleus sampling, can be employed to generate more natural-sounding text than merely sampling the most probable tokens from the output probability distribution [Holtzman et al., 2020]. Holtzman et al. [2020] have argued that natural language does not maximize probability; humans favor non-obvious language. This means that we have to determine the optimal text generation settings for Quest-GPT-2 inference to properly analyze the quality of the output quest descriptions.

We performed four mini-studies to discover the optimal text generation settings for model inference. These mini-studies were performed among the members of the game AI research group at Aalto University, and had three participants on average. We generated six to ten quest descriptions for two quests for each different text generation setting configuration with the ai-textgen Python package and Quest-GPT-2, and rated the descriptions according to their perceived quality on a 7-point Likert scale accompanied with the statement “The quest description fits the quest great.” We utilized the following setting configurations:

- Nucleus sampling with top-p values 0.5, 0.7, and 0.9
- Top-k sampling with top-k 40
- Baseline pure sampling

with or without the following additional modifiers:

- Temperature: 0.7
- Repetition penalty: 1.2

The first mini-study compared all text generation setting configurations without the additional modifiers, the second one introduced the temperature modifier, the third added in repetition penalty, and the last compared two nucleus sampling configurations, top-p values 0.5 and 0.9, to each other with both modifiers and two Likert scale statements “The quest description fits the quest great narratively” and “The quest description fits the quest great in terms of correctness.”

We observed that it is difficult to achieve a balance between the narrative quality and the correctness of details: one needs to find the text generation settings that produce an optimal degree of randomness to generate interesting yet sensible quest descriptions. In the end, we found out that nucleus sampling with top-p=0.5, temperature=0.7, and repetition penalty=1.2 produced the best results with Quest-GPT-2.

4.5 Rejecting Quest-GPT-2 Outputs

To further improve the model outputs, we implemented two simple heuristic filters that reject bad outputs. Both filters exploit the special placeholder tokens within *placeholder text*, i.e. the snake-case words that act as placeholders for unique names and numbers. For example, *character_0*, *location_0*, and *quest_giver* are such tokens in the example quest depicted in Fig. 4.4.

The first filter performs token verification: it checks that the special tokens in the output also exist in the input. For instance, the example quest (Fig. 4.4) lacks a *group_0* token that could be used as a placeholder for a faction or some other group, thus the resulting quest description should not contain said token either.

The second filter complements the first: it checks that important, user-configurable special tokens in the input are present in the output. In other words, this filter can be used to force the outputs to contain certain desired quest ingredients. For example, the descriptions for the example quest (Fig. 4.4) should include the *character_0* token, as *character_0* plays a large part in the quest.

Chapter 5

Evaluating Quest-GPT-2

Our AI model, Quest-GPT-2, is supposed to be a creative system; writing RPG quest descriptions is usually considered a creativity activity. Assessing creativity, regardless of context, is not easy: just defining *creativity* is a source of debate among (computational) creativity researchers [Guckelsberger, 2020, p. 77ff.]. It is commonly argued that a creative product must be novel and valuable [Runco and Jaeger, 2012], alongside other characteristics. Assessing the quality of our model thus comes down to measuring these two components of creativity.

Hence, we aim to find out whether players find the quest descriptions generated by Quest-GPT-2 novel and valuable. We next present our evaluation methods, show the results, and, finally, discuss the significance of the results.

5.1 Experiment Design

We performed a randomized mixed design user study. This study was conducted as an online questionnaire in which the participants were presented with quests and tasked to rate quest descriptions corresponding to the quests. For evaluation purposes, the quests were presented in a human-readable format, namely the *simple* quest metadata format (depicted in Fig. 3.1b) without the original quest descriptions.

5.2 Materials

We utilized the quest test set that we set aside during fine-tuning, as specified in Section 4.3. The test set was used because it comprises quests that were not seen by Quest-GPT-2 during its fine-tuning. They are randomly sampled from the collected professional-written RPG quests and the self-written *Minecraft* quests in equal proportions for each game presented in Table 3.1. For each quest in the test set, we generated ten quest descriptions with Quest-GPT-2, utilizing the improvements from Sections 4.4 and 4.5. Based on the 50 quests in the test set, we obtained a total of 500 quest descriptions.

The quests and their generated descriptions were embedded in an online questionnaire. To keep the workload of individual participants manageable, each participant received five quest descriptions from five randomly sampled test set quests, i.e. 25 quest descriptions in total. To counteract fatigue, the five quests were always presented along their description without interleaving the quests with each other. The presentation order of the quest descriptions within the quest "packages" was randomized to counteract order effects. Unfortunately, the order of the quest "packages" themselves could not be randomized due to the technical limitations of the employed survey platform.

5.3 Participants

We recruited the study participants online from Reddit. More specifically, we utilized various RPG sub-communities and r/SampleSize, a sub-community dedicated to (scientific) surveys. The study was advertised toward everyone aged over 18 years with RPG playing experience. We did not offer any incentives for participation.

Over the course of the study, 349 respondents participated in the study, of which 345 responses were retained. We excluded three respondents, as they only provided empty or one word answers to the free-form questions. Additionally, one respondent was excluded due to being under 18 years old. The gender break-down of participants was 71.9% male, 20.0% female, 4.9% gender variant / non-conforming, 0.6% other, and 2.6% preferred not to state their gender. 97.1% of participants stated their age: the age range of participants was 18–62 years (M=28.7, SD=8.1). The participants reported their average weekly gaming time as follows: 0.9% played less than an hour, 7.5% 1–4 hours, 15.1% 5–8 hours, 23.8% 9–12 hours, 15.7% 13–16 hours, 35.1% more than 16 hours, and 2.0% preferred not to say. Regarding the participants' familiarity with RPGs, 35.4% of participants had played *Baldur's Gate*, 30.1% *Baldur's Gate II*, 58.8% *Minecraft*, 58.6% *The Elder Scrolls IV*, 83.2% *The Elder Scrolls V*, 26.7% *Torchlight II*, 76.8% other RPGs, and 0.3% preferred not to say. When asked to mention other RPG games, the participants listed dozens of Western, Japanese, table-top inspired computer, and

MMO RPGs, reflecting the fact that most participants were avid, experienced RPG fans.

5.4 Measures

We gathered demographic and expertise data with the following questions:

1. *What is your age?* With two mutually exclusive answer options
   (a) A text box for a number
   (b) Prefer not to say
2. *What gender do you identify as?* With the mutually exclusive answer options
   (a) Male
   (b) Female
   (c) Gender variant / Non-conforming
   (d) Other
   (e) Prefer not to say
3. *How many hours per week do you play video games?* With the mutually exclusive answer options
   (a) Less than an hour
   (b) 1–4 hours
   (c) 5–8 hours
   (d) 9–12 hours
   (e) 13–16 hours
   (f) More than 16 hours
   (g) Prefer not to say
4. *Which of the following RPGs have you played?* With the freely selectable answer options
   (a) Baldur's Gate
   (b) Baldur's Gate II
   (c) Minecraft

(d) The Elder Scrolls IV: Oblivion

(e) The Elder Scrolls V: Skyrim

(f) Torchlight II

(g) Other (asks the participants to list other RPGs)

(h) None / Prefer not to say

We collected quest description ratings from the participants to assess the value of the quest descriptions. This was accomplished by accompanying each quest description with a 4-point Likert scale (Strongly Disagree – Strongly Agree) alongside the statement "I would be happy to see this quest description in a video game." An even Likert scale was chosen to divide quest descriptions into two categories: unsuitable (mean rating <2.5) and suitable (mean rating >2.5).

We moreover employed free-form questions for qualitative inquiry:

Qn 1. Which criteria did you use to assess the suitability of each quest description?

Qn 2. What upset you most about the unsuitable quest descriptions?

Qn 3. What did you like most about the suitable quest descriptions?

The first question was used to assess the criteria the participants utilized to rate the descriptions, whereas the last two ones were used to determine the strengths and weaknesses of the quest descriptions.

5.5 Procedure

Firstly, the participants were asked to read and agree to an informed consent form that was provided in the study preface. This consent form detailed the usage of the participants' personal data as required by the General Data Protection Regulation of the European Union. Secondly, the participants were asked to fill in the online questionnaire. This questionnaire began with the previously detailed demographics and expertise questions. After the demographics, the participants were shown (i) a random quest, and (ii) five different descriptions for the quest. After rating all five quest descriptions, they were shown another quest with the corresponding quest descriptions. This process was repeated five times. Finally, the participants were presented with the previously described free-form questions.

```
I am Mogrul, a money-lending merchant.
I'm not sure if you've heard of me
before, but I'm one of those people
who can lend a hand to anyone in need.
My name is Mogrul. You might know me
as Mogrul, or maybe as Drovas Relvi. I
was hired by Neloth to help him out
with a problem he had. It turned into
quite the headache for both of us.
You're going to help me deal with it.
I'll give you something called 1,000
septims. This is a new form of
currency that we're trying out here at
my location. We don't have much of it
yet, so I'm giving you a small amount
to help you get started.
If you ever need anything else, just
ask.
```

```
I'm not sure if you're aware of it,
but I've been looking for someone to
help me out with my current situation.
My name is Vex. You might know me as
Vald. I was once your lover, until we
were betrayed by our mutual friend
Mercer Frey. We had a good run
together, but now he has turned on us
both. He owes us money, so I need to
get his debtors off our backs.
If you could speak to Maven
Black-Briar about getting rid of
Vald's debts, that would be great. If
you can't do that, then I'll have to
kill him myself.
```

```
I'm sorry... I was just having a bad
day. I don't know what happened.
It all started when someone tried to
kill me. It was a Thalmor. They were
after my job and they thought I was
one of them. I had to fight them off
but it took too long. I couldn't get
away fast enough. I ran into a Dunmer
named the Khajiit. He wanted to kill
me so he came after me.
He was going to kill me anyway, but
now he doesn't want to. He says if I
go back to Windhelm, I'll die. I told
him I'd stay out of Morrowind. I
didn't tell him where I was going.
If you ever need anything, come see me.
```

(a) The worst rated quest descriptions for the three worst rated quests: "A New Debt", "Vald's Debt", and "Assassin at Large"

```
I am Officer Vai, a member of the
Flaming Fist. I'm here to talk about
something that has been plaguing us for
quite some time now.
We've had reports of bandits operating
in the area around Sword Coast for
quite some time now. They seem to be
targeting travelers and merchants
alike. We're not sure who they work
for, but we know it isn't good.
It seems as though these bandits are
working for someone else. If you could
track down the person or people behind
this, we would reward you with 50 gold
pieces per bandit scalp.
```

```
I am Jauffre, military leader here at
Cloud Ruler. I've been hearing rumors
about strange people coming into our
town.
We need to find these people before
someone else does! We don't want any
trouble with the Countess of Bruma, but
we do need to get rid of these people.
You're going to help us track down
these people. You'll go to Cloud Ruler
and speak to Steffan, one of the gate
guards. He'll tell you what he knows.
Then you'll head over to Bruma and
speak to Captain Burd, the captain of
the guard. Tell him what you found. If
there's anything suspicious, let me
know.
```

```
I've got a problem here in Arcane
University. A student named Henantier.
He was studying with me last semester,
but he got himself into too much
trouble. I'm not sure what went wrong,
but it seems like he's been doing some
very dangerous things.
He doesn't seem to care about anything
anymore. He just keeps getting into
more and more trouble. It's all my
fault. I told him to stop messing
around with magic, but he didn't
listen. Now he's stuck in a place
called Leyawiin. There's nothing I can
do for him.
If you could get Henantier out of
there, I'd give you a sizeable reward.
```

(b) The best rated quest descriptions for the three best rated quests: "Vai's Bounty Upon Bandits", "Spies", and "Through A Nightmare, Darkly"

Figure 5.1: Examples of differently rated quest descriptions

5.6 Results

Fig. 5.2 depicts box plots of averaged quest description ratings for each quest in the test set. Many of the quests had a mixture of suitable and unsuitable quest descriptions. Moreover, Fig. 5.3a shows that the median rating

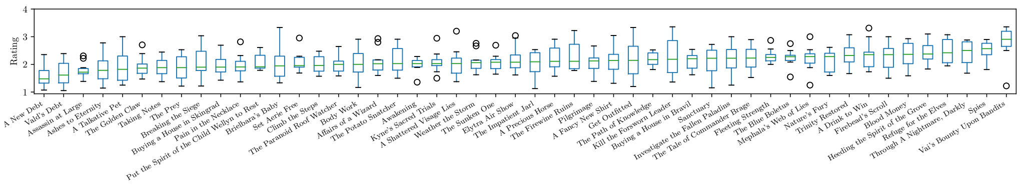

Figure 5.2: Box plots of ratings per quest description for the 50 quests in the test set, sorted by median (ascending). Each point represents the participants' mean ratings on a quest description produced for the corresponding quest

for a quest is slightly above two. When categorizing quests by their type (Fig. 5.3b) or outline length (Fig. 5.3c), the differences between the categories are small. When it comes to participants' expertise, the participants appeared more critical the more they played. However, the participants who reported playing for more than 16 hours per week, i.e. hard-core gamers, do not fit that pattern; this may be caused by positive bias.

When analyzing the answers to the free-form questions, the participants used various criteria to assess the suitability of the quest descriptions (Question 1). The most often mentioned criteria include correctness in regards to the given quest outline, internal logic as well as coherence, tone and immersiveness. Other common criteria were interestingness, the lack of repetition, grammar, narrative flow, and clear instructions. Some participants noted additional criteria, for example, humor, the length of the quest description, and the feelings that are evoked while reading the quest descriptions. There were notable differences in how the participants applied their criteria: the participants were not equally-minded about the importance of criteria, such as grammar, and a small subset of participants' answers indicate that they were lenient with their ratings, as (i) they knew that they were reading AI–generated text ("If these numbers went from 1-10 instead of 1-4, I think they'd get the same ratings, for the most part"), (ii) they were not native English speakers ("note: I'm not native speaker"), or (iii) they appreciated the unintentional humor often found within computer-generated text ("They [suitable descriptions] were humorous at times").

The participants' opinions on unsuitable (Question 2) and suitable (Question 3) quest descriptions echoed their assessment criteria: the unsuitable quest descriptions failed the criteria, whereas the suitable ones fulfilled them. In greater detail, the unsuitable descriptions were thought to be non-sensible or illogical, contained unnecessary details, repetition and conflicting information, had poor grammar to the point of "reading 'off' as if poorly trans-

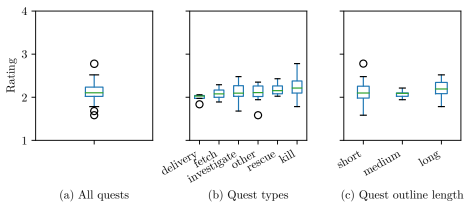

Figure 5.3: Box plots of ratings per different quest-related factors. One datapoint represents the mean rating for a quest in the test set. Fig. 5.3b is sorted by median (ascending)

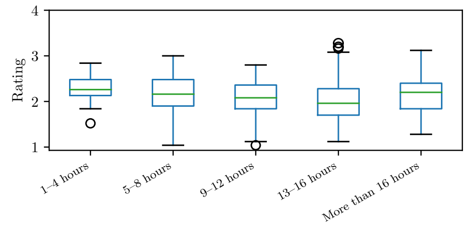

Figure 5.4: Box plots of averaged ratings per participant, grouped by participants' average weekly playtime (groups holding <5% participants were omitted)

lated from a Chinese comic", or were simply boring lists of facts. On the contrary, the suitable descriptions were found clear, surprising, fun, original, and believable even to the point of seeming human-authored. However, some participants noted that there were no suitable quest descriptions, supporting the notion that the descriptions vary greatly in quality.

Fig. 5.1 shows examples of both badly and well rated quest descriptions. In addition to highlighting many of the participants' thoughts on unsuitable quest descriptions, the badly rated descriptions indicate that Quest-GPT-2 cannot reliably discern different entities from each other even if unique names are substituted with generic placeholders. This behavior is likely inherent to GPT-2, and made worse with complicated relationships between different characters. For instance, Mogrul, the quest-giver of "A New Debt" (the quest depicted in Fig. 5.5a), and Drovas Relvi, Mogrul's debtor in the same quest, are supposed to be different people, yet in the top-most quest description in Fig. 5.1a the quest-giver states that "My name is Mogrul. You might know me as Mogrul, or maybe as Drovas Relvi."

5.7 Discussion

In its current form, Quest-GPT-2 cannot be used to autonomously generate high-quality quest descriptions reliably. Our results indicate that the underlying GPT-2 model lacks the ability (i) to distinguish between multiple entities as well as handle them accordingly, and (ii) to "glue" quest ingredients together in a satisfactory manner while also not relaying illogical information. On the contrary, GPT-3 has been shown to offer vast, general

```
This is an RPG quest from The
Elder Scrolls V: Skyrim.
The quest-giver is called
Mogrul. Mogrul is a
money-lender.
The quest-giver gives a quest
to the player. The player's
objective is to help Drovas
Relvi with his debt to Mogrul.
The player should first pay
Drovas Relvi's debt to
complete their objective.
The player will receive the
following rewards for
completing the quest
objective: Mogrul will stop
sending his debt collectors
after the player.
The following characters are
related to this quest: Drovas
Relvi (a man the player
recruited for Neloth, and thus
became untouchable), Neloth (a
powerful male wizard).
The following items are
related to this quest: 1,000
septims (a currency).
There are some important facts
concerning this quest. Drovas
Relvi owes Mogrul 1,000
septims: Mogrul thinks that
the player should pay Drovas
Relvi's debt.
This is the quest description,
the quest-giver explaining the
quest to the player:
"
```

(a) Input quest outline in the *narrative* format with *raw text*

```
I hear you've been helping out Neloth. I'm not
surprised. Neloth isn't very good at repaying his
debts. I hear you helped out Drovas Relvi, too.
Drovas Relvi owes me 1,000 septims. I'd like for
you to pay his debt. Then I'll stop sending my
debt collectors after you.
```

```
I'm sure you're wondering what you're doing here.
Well, I need your help.
Drovas Relvi owes me 1,000 septims, but I can't
go after him myself. He works for Neloth, you
see. If I attack him, Neloth will know.
I'm sure you can understand. I'm willing to pay
you 500 gold to help me. I will follow you to
Relvi's house. You just have to deliver the money.
My associate will take care of the rest.
```

```
Fellow, what's the word on the street?
Drovas Relvi owes me 1,000 septims. That can't be
good for his health. Has he talked to you?
I don't care what you get him to do. Just get my
money.
```

```
There's a fellow in town who owes me a fair bit
of coin. Drovas Relvi. I'd like to collect, but
he's Neloth's friend and I don't want to make
trouble.
I'm trying to help him out, but I'm not sure how.
If you could talk to him, maybe you could
convince him to pay me back.
If you do that, I'll make sure my collectors
don't bother you again.
```

(b) Four random output quest descriptions generated with GPT-3

Figure 5.5: A quest generation demo with the quest "A New Debt" and GPT-3 (OpenAI API Playground, default text generation settings apart from response length of 700). The quests "Ashes to Eternity", "Assassin at Large", and "Vald's Debt" were given as examples to GPT-3 beforehand.

improvements in text quality over GPT-2 [Brown et al., 2020], as implied in Section 2.3. GPT-3 would likely handle these two aspects of quest description generation better as well. As a demonstration, Fig. 5.5 illustrates a quest generation demo with not fine-tuned GPT-3 and the quest with the worst rated descriptions, i.e. "A New Debt". In comparison to Quest-GPT-2, the descriptions outputted by GPT-3 are noticeably more coherent than the worst rated Quest-GPT-2 descriptions depicted in Fig. 5.1a. GPT-3 was recently made fine-tunable on the OpenAI API: fine-tuning it would likely result in even higher-quality quest descriptions, especially with additional enhancements like *placeholder text*.

However, there are some realistic use cases for our current model. Firstly, many of the poorly rated quest descriptions outputted by Quest-GPT-2 only contain few issues, or even just individual illogical sentences. Therefore, the model could be used as an assistant in quest writing. For instance, a professional RPG writer could first give a rough, simplified quest outline to Quest-GPT-2, and then fill in more complex details into the generated output themselves. Secondly, Quest-GPT-2 could be used to generate quest ideas: one can supply the starting sentence of a quest outline to generate the rest of the outline and the quest description. Lastly, there were quest descriptions that were deemed suitable by people. We calculated the curation coefficient, i.e. the ratio of human-acceptable outputs from any given creative system [Colton and Wiggins, 2012]: the curation coefficient of 0.22 implies that roughly one in five quest descriptions will be acceptable by people. Hence, Quest-GPT-2 could be used offline to generate quest descriptions which can, after only little human curation, be used in a video game without further changes.

There are some limitations to our study. Firstly, there are biases in the participants' attitudes toward AI-generated text: both positive and negative bias were observed. The former was evident from the participants using lenient ratings as described previously, and the latter was observed from e.g. one of the participants describing bad experiences with procedurally generated quests from *The Elder Scrolls V: Skyrim*. Biases are often present, when people judge computer-generated creative artifacts: Colton and Wiggins [2012] have stated that

> there is a natural predilection for people to attribute creativity to human programmers, users and audiences instead of software and hardware. It seems that people allow their beliefs that machines can't possibly be creative to bias their judgement on such issues,

indicating that both lenient and critical participants question the creativity of Quest-GPT-2. To alleviate bias, one could use a mixture of human-written and AI-generated quest descriptions instead of only AI-generated descriptions. Secondly, our quest data set is focused on AI games with medievalesque fantasy settings, thus generalizing the results of the study to other types of settings is not feasible.

On a general note, it seems that there is no objective consensus for what makes up a good quest description: some study participants preferred short, no-nonsense descriptions without unnecessary details, whereas others liked longer descriptions laced with in-game lore. Regarding quest objectives, there were participants who would rather only receive hints about what to do, and others who preferred in-dept instructions.

Chapter 6

Conclusions and Future Work

In this thesis, we have investigated using a modern language model, GPT-2, to autonomously generate quest descriptions for 3D games. We built a novel quest data set, and developed a strategy for learning from a limited training data by substituting proper nouns and numbers with placeholders. We fine-tuned GPT-2 into quest description generating Quest-GPT-2, and conducted an online user study to evaluate the quest descriptions generated with it.

Our results are encouraging, yet the quality of the generated descriptions varied greatly. Despite our name substitution strategy, Quest-GPT-2 often makes mistakes related to handling a large number of entities, such as characters, groups, and locations. Moreover, Quest-GPT-2 often generates descriptions with questionable logic, repetition, poor grammar, and unnecessary information. While using our model automatically and online is not yet viable, we propose several means on how Quest-GPT-2 can already be used by designers offline.

We are confident that the next generation of language models could be fine-tuned with our quest data set to alleviate the discussed issues. For example, our quest generation demo with not fine-tuned GPT-3 implies that it can generate more coherent quest descriptions than Quest-GPT-2. Other potential areas of future work are personalizing quest descriptions for different kinds of RPG players and player characters, replacing our simple heuristic filters with an AI critic for rejecting dissatisfying model outputs as well as using grammar checking tools or other algorithms for improving text quality, and generating other quest-related artifacts, e.g. quest names, journal entries and dialogue trees, in addition to quest descriptions.

Bibliography

Agafonova, Y., Tikhonov, A., and Yamshchikov, I. P. (2020). Paranoid transformer: Reading narrative of madness as computational approach to creativity. *Future Internet*, 12(11):182.

Ammanabrolu, P., Broniec, W., Mueller, A., Paul, J., and Riedl, M. (2019). Toward automated quest generation in text-adventure games. In *Proceedings of the Workshop on Computational Creativity in Language Generation*, pages 1–12.

Bethesda Game Studios (2006). *The Elder Scrolls IV: Oblivion*. Game [PC]. Bethesda Softworks, Rockville, Maryland, United States.

Bethesda Game Studios (2011). *The Elder Scrolls V: Skyrim*. Game [PC]. Bethesda Softworks, Rockville, Maryland, United States.

BioWare (1998). *Baldur's Gate*. Game [PC]. Interplay Entertainment, Los Angeles, California, United States.

BioWare (2000). *Baldur's Gate II: Shadows of Amn*. Game [PC]. Interplay Entertainment, Los Angeles, California, United States.

Blizzard Entertainment (2004). *World of Warcraft*. Game [PC]. Blizzard Entertainment, Irvine, California, United States.

Boden, M. A. (1990). *The Creative Mind: Myths and Mechanisms*. Weidenfeld & Nicolson, London, United Kingdom, 1st edition.

Breault, V., Ouellet, S., and Davies, J. (2021). Let CONAN tell you a story: Procedural quest generation. *Entertainment Computing*, 38(3):100422.

Brockman, G., Murati, M., Welinder, P., and OpenAI (2020). OpenAI API. https://openai.com/blog/openai-api/. Last accessed November 2020.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. arXiv:2005.14165.

Calvin, A. and Michael, N. (2007). The quest in a generated world. In *Proceedings of the DiGRA International Conference: Situated Play*, pages 503–509.

Colton, S. and Wiggins, G. (2012). Computational creativity: The final frontier? In *European Conference on Artificial Intelligence (ECAI)*, pages 21–26.

Cook, M. and Colton, S. (2014). Ludus ex machina: Building a 3D game designer that competes alongside humans. In *Proceedings of the International Conference on Computational Creativity*, pages 54–62.

Cook, M., Colton, S., and Gow, J. (2016a). The ANGELINA videogame design system–part I. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(2):192–203.

Cook, M., Colton, S., and Gow, J. (2016b). The ANGELINA videogame design system–part II. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(3):254–266.

Cook, M. and Smith, G. (2015). Formalizing non-formalism: Breaking the rules of automated game design. In *Proceedings of the Conference on the Foundations of Digital Games*, pages 1–5.

Doran, J. and Parberry, I. (2011). A prototype quest generator based on a structural analysis of quests from four MMORPGs. In *Proceedings of the international workshop on procedural content generation in games*, pages 1–8.

Freiknecht, J. and Effelsberg, W. (2020). Procedural generation of interactive stories using language models. In *International Conference on the Foundations of Digital Games*, pages 1–8.

Gervás, P. (2009). Computational approaches to storytelling and creativity. *AI Magazine*, 30(3):49–62.

Guckelsberger, C. (2020). *Intrinsic Motivation in Computational Creativity Applied to Videogames*. PhD thesis, School of Electrical Engineering and Computer Science, Queen Mary, University of London. 306 pages.

He, C. (2020). Understand the pricing of GPT3. https://medium.com/@chengh/understand-the-pricing-of-gpt3-e646b2d63320. Last accessed November 2020.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration. arXiv:1904.09751.

Köbis, N. and Mossink, L. D. (2021). Artificial intelligence versus Maya Angelou: Experimental evidence that people cannot differentiate AI-generated from human-written poetry. *Computers in Human Behavior*, 114(1):106553.

Lamb, C., Brown, D. G., and Clarke, C. L. A. (2018). Evaluating computational creativity: An interdisciplinary tutorial. *ACM Computing Surveys (CSUR)*, 51(2):1–34.

Lee, H. H., Shu, K., Achananuparp, P., Prasetyo, P. K., Liu, Y., Lim, E.-P., and Varshney, L. R. (2020). RecipeGPT: Generative pre-training based cooking recipe generation and evaluation system. In *Companion Proceedings of the Web Conference*, pages 181–184.

Lee, J.-S. (2020). Controlling patent text generation by structural metadata. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, pages 3241–3244.

Liapis, A., Yannakakis, G. N., and Togelius, J. (2014). Computational game creativity. In *International Conference on Computational Creativity*, pages 285–292.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating Wikipedia by summarizing long sequences. arXiv:1801.10198.

Mojang Studios (2011). *Minecraft*. Game [PC]. Mojang Studios, Stockholm, Sweden.

Pérez y Pérez, R. (2018). The computational creativity continuum. In *Proceedings of the International Conference on Computational Creativity*, pages 177–184.

Pita, J., Magerko, B., and Brodie, S. (2007). True story: Dynamically generated, contextually linked quests in persistent systems. In *Proceedings of the Conference on Future Play*, pages 145–151.

Propp, V. I. (1968). *Morphology of the Folktale.* University of Texas Press, Austin, United States, 2nd edition.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI Blog.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog.

Runco, M. A. and Jaeger, G. J. (2012). The standard definition of creativity. *Creativity Research Journal*, 24(1):92–96.

Runic Games (2012). *Torchlight II.* Game [PC]. Runic Games, Seattle, Washington, United States.

Soares de Lima, E., Feijó, B., and Furtado, A. L. (2019). Procedural generation of quests for games using genetic algorithms and automated planning. In *Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pages 144–153.

Stocker, A. and Alvin, C. (2018). Non-linear quest generation. In *Proceedings of the International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 213–216.

Togelius, J., Yannakakis, G. N., Stanley, K. O., and Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186.

van Stegeren, J. and Theune, M. (2020). Fantastic strings and where to find them: The quest for high-quality video game text corpora. In *Proceedings of the Workshop on Intelligent Narrative Technologies*, pages 1–8.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. arXiv:1706.03762.

Veale, T., Amílcar Cardoso, F., and Pérez y Pérez, R. (2019). Systematizing creativity: A computational view. In *Computational Creativity*, pages 1–19.

Ventura, D. (2016). Beyond computational intelligence to computational creativity in games. In *IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. arXiv:1909.08593.

Appendix A

JSON Representation for Quests

```
{
        "name": "the name of the quest",
        "objective": "quest objective",
        "first_tasks": ["a list of tasks that should be done to fulfill the
        objective"],
        "first_task_locations": ["a list of locations correspondind with
        the tasks, similar to the locations field"],
        "quest_giver": {
                "name": "the name or title of the quest giver",
                "description": "a brief, archetypal description of the
                quest giver",
                "location": "the whereabouts of the quest giver"
        },
        "reward": [a list rewards, a reward is defined {
                "name": "the name of the reward",
                "description": "a brief, common description of the reward",
                "amount": the number of received rewards
        }],
        "characters": [(optional) a list of related characters, a character
        is defined similarly to the quest giver],
        "enemies": [(optional) a list of related groups of enemies, mostly
        used for declaring a set number of enemies for a quest, a group of
        enemies is defined similarly to a reward],
        "items": [(optional) a list of related items, e.g tangible items,
        or even some more abstract ones like rituals, an item is defined
        similarly to a reward],
        "groups": [(optional) a list of related groups, e.g. factions,
        races, or creatures, where a group is defined {
                "name": "the name of the group",
                "description": "a brief, common description of the group"
        }],
        "locations": [(optional) a list of related locations, where a
        location is defined {
                "name": "the name of the location",
                "description": "a brief, common description of the location"
        }],
        "tools": ["important facts related to the quest"],
        "description": "the quest description"
}
```